TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA TOÁN – CƠ – TIN HỌC


# TÊN ĐỀ TÀI:
# DỰ ĐOÁN MỨC THU NHẬP NGƯỜI TRƯỞNG THÀNH THÔNG QUA DỮ LIỆU ĐIỀU TRA DÂN SỐ HOA KỲ

## BÀI TIỂU LUẬN KẾT THÚC HỌC PHẦN

**Học phần: Mining Big Datasets**

**Giảng viên hướng dẫn: TS. Lê Hồng Phương**

**Thực hiện: Phạm Quang Hiếu**

# MỤC LỤC

# DANH MỤC HÌNH ẢNH

## GIỚI THIỆU VÀ MỤC TIÊU BÀI TOÁN

Kể từ năm 1790, c ứ 10 năm một lần, theo yêu cầu của Hiến pháp Hoa Kỳ, Cục Điều tra Dân số sẽ tiến hành kiểm tra tổng số người dân ở Hoa Kỳ và hỏi họ những câu hỏi để giúp tìm hiểu thêm về đất nước nói chung: chúng ta là ai, chúng ta sống ở đâu, chúng ta là gì, chúng ta làm nghề gì, mức tiền kiếm được ,tình trạng hôn nhân, tình trạng cư trú , và nhiều chủ đề khác. Dữ liệu thu thập được sử dụng để phân bổ các ghế trong Quốc hội, phân phối viện trợ liên bang, xác định các khu vực lập pháp, giúp chính quyền liên bang, tiểu bang và địa phương đo lường nhu cầu về nhà ở, dự đoán nhu cầu trong tương lai và xác định xu hướng.

Tập dữ liệu này chứa dữ liệu điều tra dân số có trọng số được trích xuất từ các cuộc điều tra dân số hiện tại năm 1994 và 1995 do Cục điều tra dân số Hoa Kỳ (http://www.census.gov/) thực hiện.Dữ liệu chứa 42 biến liên quan đến nhân khẩu học và việc làm, mục tiêu là dự đoán mức thu nhập của người trưởng thành.

### 1. Xác định tập dữ liệu

Dữ liệu trong dự án này được tải xuống từ kh o lưu trữ máy học UCI ( UCI Machine Learning Repository).

https://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29

#### *1.1. Xác định các biến trong bộ dữ liệu*

Bộ dữ liệu là bảng điều tra dân số trong khoảng thời gian từ năm 1994 tới năm 1995 bởi Cục Dân số Hoa Kỳ, gồm 42 thông số:

- Income: là biến cần dự báo. Dựa vào các biến biết trước, ta cần xác định thu nhập hàng năm của một người cao hơn hay thấp hơn 50.000$.
- Age : Tuổi tác. Độ tuổi của những người được khảo sát từ 0 – 90 tuổi.
- Class of worker: Lớp người lao động – xác định đối tượng thuộc chính phủ, hay kinh doanh, chưa đi làm, hoặc không được trả lương.
- Industry code: Mã ngành đang làm việc, có tổng cộng 52 mã ngành (0 – 51) trong bộ dữ liệu.
- Occupation code: Mã công việc, có tổng cộng 47 mã công việc (0- 46) trong bộ dữ liệu
- Education : Trình độ học vấn.
- Wage per hour: Mức lương theo giờ.
- Enrolled in edu inst last wk: Chứa thông tin cấp học của đối tượng vừa mới được tuyển. Đa phần thông tin ở dạng Not in Universal.
- Marital status: Tình trạng hôn nhân.

- Major industry code: Diễn giải bằng ngôn ngữ cho mã số ở cột industry code. Cột này có thể được nội suy ra từ cột industry code.
- Major occupation code : Diễn giải bằng ngôn ngữ cho mã số ở cột occupation code.
- Mace: chủng tộc
- Hispanic Origin: Nếu quan sát có nguồn gốc Tây Ban Nha, cột này sẽ thể hiện nguồn gốc, nếu không sẽ có giá trị All other.
- Sex: giới tính
- Member of a labor union: Có phải thành viên của một liên đoàn lao động hay không, nhận các giá trị Yes hoặc No
- Reason for unemployment: Lý do thất nghiệp bị sa thải, xin nghỉ việc hoặc mới vào nghề.
- Full or part time employment stat : Tính chất công việc toàn thời gian hay bán thời gian,…
- Capital gains: Lợi nhuận từ đầu tư.
- Capital lostes: Khoản thua lỗ từ đầu tư
- Divdends from stocks: cổ tức
- Federal income tax liability: Nghĩa vụ đóng thuế - mỗi cá nhân tuỳ vào độ tuổi, tình trạng gia đình sẽ có nghĩa vụ thuế khác nhau.
- Tax filer status: Khu vực mà quan sát đó nộp thuế.
- Region of previous residence: Nơi ở trước của quan sát – nước ngoài hay một bang tại Mỹ.
- State of previous residence: Trạng thái cư trú trước đây – Có sở hữu nhà hay không, là chủ nhà hay chỉ là thành viên, là trẻ em hay người lớn, …
- Detailed household and family stat: Thông tin về gia đình.
- Detailed household summary in household: Nhận các giá trị số thực từ 43 đến 16300
- Instance weight: Biểu diễn số thứ tự của quan sát có được từ lấy mẫu phân tầng. Để thực hiện thống kê thực tế và đưa ra kết luận, trường này là cần thiết nhưng trong bài toán phân loại thì không dùng đến.
- Migration code-change in msa, migration code-change in reg, migration code-move within reg : Sự di cư giữa các vùng của quan sát.
- Live in this house 1 year ago: nhận 2 giá trị Yes – No.
- Family members under 18: Số thành viên dưới 18 tuổi, nhận giá trị từ 0 – 6
- Total person earnings: Thành viên nào trong gia đình đem lại thu nhập.
- Country of birth father, country of birth mother, country of birth self: Nơi sinh của cha, mẹ, bản thân.
- Citizenship: Người nước ngoài hay bản địa, nơi sinh và nơi ở hiện tại.
- Total person income: Số người có thu nhập, nhận giá trị 0,1 hoặc 2

- Own business or self employed: tự kinh doanh – nhận giá trị Yes hoặc No
- Taxable income amount: Số thu nhập chịu thuế, nhận giá trị 0,1 hoặc 2
- Hours per week: số giờ làm việc mỗi tuần, nhận giá trị từ 0 – 52.
- Veterans benefits: phúc lợi cho cựu chiến binh, nhận 2 giá trị 94 – 95.

### *1.2. Khai phá và xử lý dữ liệu missing*

```
RangeIndex: 199523 entries, 0 to 199522
Data columns (total 42 columns):

root
 |-- age: integer (nullable = true)
 |-- class of worker: string (nullable = true)
 |-- industry code: integer (nullable = true)
 |-- occupation code: integer (nullable = true)
 |-- education: string (nullable = true)
 |-- wage per hour: integer (nullable = true)
 |-- enrolled in edu inst last wk: string (nullable = true)
 |-- marital status: string (nullable = true)
 |-- major industry code: string (nullable = true)
 |-- major occupation code: string (nullable = true)
 |-- mace: string (nullable = true)
 |-- hispanic Origin: string (nullable = true)
 |-- sex: string (nullable = true)
 |-- member of a labor union: string (nullable = true)
 |-- reason for unemployment: string (nullable = true)
 |-- full or part time employment stat: string (nullable = true)
 |-- capital gains: integer (nullable = true)
 |-- capital losses: integer (nullable = true)
 |-- divdends from stocks: integer (nullable = true)
 |-- federal income tax liability: string (nullable = true)
 |-- tax filer status: string (nullable = true)
 |-- region of previous residence: string (nullable = true)
 |-- state of previous residence: string (nullable = true)
 |-- detailed household and family stat: string (nullable = true)
 |-- detailed household summary in household: double (nullable = true)
 |-- instance weight: string (nullable = true)
 |-- migration code-change in msa: string (nullable = true)
 |-- migration code-change in reg: string (nullable = true)
 |-- migration code-move within reg: string (nullable = true)
 |-- live in this house 1 year ago: string (nullable = true)
 |-- family members under 18: integer (nullable = true)
 |-- total person earnings: string (nullable = true)
 |-- country of birth father: string (nullable = true)
 |-- country of birth mother: string (nullable = true)
 |-- country of birth self: string (nullable = true)
 |-- citizenship: string (nullable = true)
 |-- total person income: integer (nullable = true)
 |-- own business or self employed: string (nullable = true)
 |-- taxable income amount: integer (nullable = true)
 |-- hours per week: integer (nullable = true)
 |-- veterans benefits: integer (nullable = true)
 |-- income: string (nullable = true)
```

*Hình 1 – Tổng quan về bộ dữ liệu*

**Nhận xét**

Bộ dữ liệu gồm 199.523 quan sát với 42 biến: Trong đó có 29 biến có kiểu dữ liệu *string*, 12 biến có kiểu dữ liệu *integer* và 1 biến có kiến dữ liệu là *double*.

Do mục tiêu của bài toán là dự đoán mức thu nhập (*income*) của người trưởng thành tại nước Mỹ. Trong khi dữ liệu khảo sát bao gồm tất cả các độ tuổi do vậy để tối ưu hóa mô hình, chúng ta sẽ loại bỏ đi các khảo sát có độ tuổi nhỏ hơn 18 tuổi. Bộ dữ liệu sau khi đã lược bỏ ta còn lại 143.531 quan sát.

```
adult_df = df.filter(df['age'] >= 18)
print("Number of row: {}".format(adult_df.count()))
```

```
Number of row: 143531
```

### *1.2.1. Kiểm tra các quan sát đầu trong bộ dữ liệu*

```
+------+--------------------+-------------+---------------+--------------------+------------+--------------------------+
|income|     class of worker|industry code|occupation code|           education|wage per hour|enrolled in edu inst last wk|
+------+--------------------+-------------+---------------+--------------------+------------+--------------------------+
|-50000|     Not in universe|            0|              0| High school grad...|           0|             Not in universe|
|-50000| Self-employed-no...|            4|             34| Some college but...|           0|             Not in universe|
|-50000|     Not in universe|            0|              0|          10th grade|           0|                High school|
|-50000|     Not in universe|            0|              0|            Children|           0|             Not in universe|
|-50000|     Not in universe|            0|              0|            Children|           0|             Not in universe|
|-50000|             Private|           40|             10| Some college but...|        1200|             Not in universe|
|-50000|             Private|           34|              3| Bachelors degree...|           0|             Not in universe|
|-50000|             Private|            4|             40| High school grad...|           0|             Not in universe|
|-50000|    Local government|           43|             26| Some college but...|         876|             Not in universe|
|-50000|             Private|            4|             37| Some college but...|           0|             Not in universe|
+------+--------------------+-------------+---------------+--------------------+------------+--------------------------+
```

```
+--------------------+--------------------+--------------------+--------------+-------+----------------------+
|      marital status| major industry code|                mace|hispanic Origin|    sex|member of a labor union|
+--------------------+--------------------+--------------------+--------------+-------+----------------------+
|             Widowed| Not in universe ...|               White|     All other| Female|        Not in universe|
|            Divorced|        Construction|               White|     All other|   Male|        Not in universe|
|       Never married| Not in universe ...| Asian or Pacific...|     All other| Female|        Not in universe|
|       Never married| Not in universe ...|               White|     All other| Female|        Not in universe|
|       Never married| Not in universe ...|               White|     All other| Female|        Not in universe|
| Married-civilian...|       Entertainment| Amer Indian Aleu...|     All other| Female|                    No|
| Married-civilian...| Finance insuranc...|               White|     All other|   Male|        Not in universe|
|       Never married|        Construction|               White|     All other| Female|        Not in universe|
| Married-civilian...|           Education|               White|     All other| Female|                    No|
| Married-civilian...|        Construction|               White|     All other|   Male|        Not in universe|
+--------------------+--------------------+--------------------+--------------+-------+----------------------+
```

| reason for unemployment | full or part time employment stat | capital gains | capital losses | divdends from stocks |
|---|---|---|---|---|
| Not in universe | Not in labor force | 0 | 0 | 0 |
| Not in universe | Children or Arme... | 0 | 0 | 0 |
| Not in universe | Not in labor force | 0 | 0 | 0 |
| Not in universe | Children or Arme... | 0 | 0 | 0 |
| Not in universe | Children or Arme... | 0 | 0 | 0 |
| Not in universe | Full-time schedules | 0 | 0 | 0 |
| Not in universe | Children or Arme... | 5178 | 0 | 0 |
| Job loser - on l... | Unemployed full-... | 0 | 0 | 0 |
| Not in universe | Full-time schedules | 0 | 0 | 0 |
| Not in universe | Children or Arme... | 0 | 0 | 0 |

| federal income tax liability | tax filer status | region of previous residence | state of previous residence |
|---|---|---|---|
| Nonfiler | Not in universe | Not in universe | Other Rel 18+ ev... |
| Head of household | South | Arkansas | Householder |
| Nonfiler | Not in universe | Not in universe | Child 18+ never ... |
| Nonfiler | Not in universe | Not in universe | Child <18 never ... |
| Nonfiler | Not in universe | Not in universe | Child <18 never ... |
| Joint both under 65 | Not in universe | Not in universe | Spouse of househ... |
| Joint both under 65 | Not in universe | Not in universe | Householder |
| Single | Not in universe | Not in universe | Secondary indivi... |
| Joint both under 65 | Not in universe | Not in universe | Spouse of househ... |
| Joint both under 65 | Not in universe | Not in universe | Householder |

| detailed household and family stat | migration code-change in msa | migration code-change in reg | migration code-move within reg |
|---|---|---|---|
| Other relative o... | ? | ? | Not in universe ... |
| Householder | Same county | Same county | No |
| Child 18 or older | ? | ? | Not in universe ... |
| Child under 18 n... | Nonmover | Nonmover | Yes |
| Child under 18 n... | Nonmover | Nonmover | Yes |
| Spouse of househ... | ? | ? | Not in universe ... |
| Householder | Nonmover | Nonmover | Yes |
| Nonrelative of h... | ? | ? | Not in universe ... |
| Spouse of househ... | ? | ? | Not in universe ... |
| Householder | Nonmover | Nonmover | Yes |

| live in this house 1 year ago | instance weight | family members under 18 | total person earnings | country of birth father |
|---|---|---|---|---|
| ? | ? | 0 | Not in universe | United-States |
| Yes | MSA to MSA | 1 | Not in universe | United-States |
| ? | ? | 0 | Not in universe | Vietnam |
| Not in universe | Nonmover | 0 | Both parents pre... | United-States |
| Not in universe | Nonmover | 0 | Both parents pre... | United-States |
| ? | ? | 1 | Not in universe | Philippines |
| Not in universe | Nonmover | 6 | Not in universe | United-States |
| ? | ? | 4 | Not in universe | United-States |
| ? | ? | 5 | Not in universe | United-States |
| Not in universe | Nonmover | 6 | Not in universe | United-States |

| country of birth mother | country of birth self | citizenship | total person income |
|---|---|---|---|
| United-States | United-States | Native- Born in ... | 0 |
| United-States | United-States | Native- Born in ... | 0 |
| Vietnam | Vietnam | Foreign born- No... | 0 |
| United-States | United-States | Native- Born in ... | 0 |
| United-States | United-States | Native- Born in ... | 0 |
| United-States | United-States | Native- Born in ... | 2 |
| United-States | United-States | Native- Born in ... | 0 |
| United-States | United-States | Native- Born in ... | 0 |
| United-States | United-States | Native- Born in ... | 0 |
| United-States | United-States | Native- Born in ... | 0 |

| own business or self employed | taxable income amount | hours per week | veterans benefits |
|---|---|---|---|
| Not in universe | 2 | 0 | 95 |
| Not in universe | 2 | 52 | 94 |
| Not in universe | 2 | 0 | 95 |
| Not in universe | 0 | 0 | 94 |
| Not in universe | 0 | 0 | 94 |
| Not in universe | 2 | 52 | 95 |
| Not in universe | 2 | 52 | 94 |
| Not in universe | 2 | 30 | 95 |
| Not in universe | 2 | 52 | 95 |
| Not in universe | 2 | 52 | 94 |

only showing top 10 rows

*Hình 2 – Kiểm tra các quan sát đầu trong bộ dữ liệu*

*** Nhận xét:**

Tổng quan bộ dữ liệu không chứa các giá trị bị thiếu (null) tuy nhiên trong các trường dữ liệu có chứa các dữ liệu dạng *'Not in universe' và '?'*. Các biến trên có thể coi là các dữ liệu thiếu trong quá trình thu thập. Do vậy ta có thể coi đó là các dữ liệu null.

**Hướng xử lý:** Ta sẽ thay thế các dữ liệu *'Not in universe' và '?'* thành các dữ liệu rỗng để xử lý.

### *1.2.2. Kiểm tra và xử lý các giá trị null (rỗng), dữ liệu không xác định trong bộ dữ liệu*

```
count_null = adult_df.select([count(when(isnan(c) | col(c).isNull(), c)).alias(c) for c in adult_df.columns])
for i in count_null.columns:
    if count_null.select(i).collect()[0][0] != 0:
        print('Number null value in {}: {}'.format(i,count_null.select(i).collect()[0][0]))
```

```
Number null value in class of worker: 46996
Number null value in enrolled in edu inst last wk: 136243
Number null value in major occupation code: 47189
Number null value in member of a labor union: 124912
Number null value in reason for unemployment: 137993
Number null value in tax filer status: 132083
Number null value in region of previous residence: 132617
Number null value in instance weight: 71811
Number null value in migration code-change in msa: 71811
Number null value in migration code-change in reg: 71811
Number null value in live in this house 1 year ago: 132083
Number null value in total person earnings: 143531
Number null value in country of birth father: 5538
Number null value in country of birth mother: 5003
Number null value in country of birth self: 2957
Number null value in own business or self employed: 141550
```

*Hình 3 –Các trường dữ liệu có chứa các giá trị null (rỗng) trong bộ dữ liệu*

*** Nhận xét:**

Các giá trị rỗng sẽ làm giảm độ chính xác khi đo đếm và vẽ sơ đồ cho dữ liệu. Qua kiểm tra ta thấy giá trị null nằm trong các trường trên Hình 3. Chi tiết ta có:

Các cột:

+ "enrolled in edu inst last wk"
+ "member of a labor union"

```
+"reason for unemployment"
+"tax filer status"
+ "region of previous residence"
+ "live in this house 1 year ago"
+ "total person earnings"
+ "own business or self employed"
```

chứa nhiều giá trị rỗng (trên 90%). Do đó sự có mặt của các biến trên trong mô hình có thể làm giảm độ chính xác của mô hình.

**Hướng xử lý:**Loại bỏ các biến trênkhỏi mô hình.

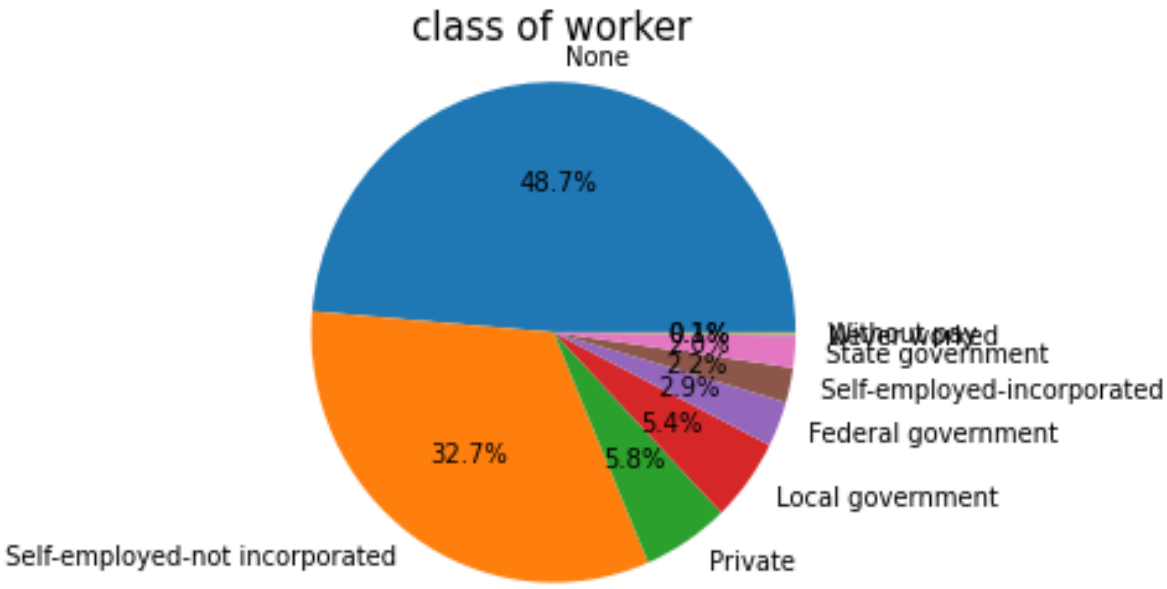
```
df_Columns=["enrolled in edu inst last wk",
            "member of a labor union",
            "reason for unemployment",
            "tax filer status",
            'region of previous residence',
            'live in this house 1 year ago',
            'total person earnings',
            'own business or self employed'
           ]
```

```
adult_df = adult_df.drop(*df_Columns)
print("Number of collumns: {}".format(len(adult_df.columns)))
```

```
Number of collumns: 34
```

Sau khi đã loại bỏ các biến chứa nhiều giá trị null, bộ dữ liệu còn lại 34 biến.

Cột *"class of worker"* chứa 71.811 giá trị rỗng. Nhận thấy số lượng số liệu rỗng có tỉ lệ tương đương với trường *"Seft-employed-not incoporated"* trong biến trên. Do đó có thể các dữ liệu *Not in universe* trong biến này vẫn có ý nghĩa thống kê.




*Hình 4 – Phân bố giá trị trong biến class of worker*

**Hướng xử lý:** Vẫn giữ lại các giá trị None (Not in universe) coi như 1 giá trị trong biến

Ta làm tương tự với các biến có tỉ lệ giá trị null tương tự như biến *"class of worker" : 'major occupation code', 'instance weight', 'migration code -change in msa', 'migration code-change in reg'.*

Các cột: *"country of birth father", "country of birth mother", "country of birth self", "own business or self employed"* số lượng biến null chiếm tỉ lệ nhỏ (dưới 3%).

**Hướng xử lý:** loại bỏ các dòng dữ liệu có chứa dữ liệu null trong bộ dữ liệu.

*** Nhận xét:**

- Sau quá trình xử lý dữ liệu null ta thu được bộ dữ liệu còn lại gồm 34 biến và 136.826 điểm dữ liệu.

```
adult_df = adult_df.na.drop()
```

```
print("Number of row: {}".format(adult_df.count()))
```

```
Number of row: 136826
```

```
print("Number of columns: {}".format(len(adult_df.columns)))
```

```
Number of columns: 34
```

## 2. Phân tích sự tương quan giữa biến giải thích (x) và biến phụ thuộc (y)

### *2.1.Tương quan giữa biến income và các biến định lượng*

```
def corr(df):
    vector_col = "corr_features"
    assembler = VectorAssembler(inputCols=df.columns, outputCol=vector_col)
    df_vector = assembler.transform(df).select(vector_col)

    # get correlation matrix
    matrix = Correlation.corr(df_vector, vector_col)
    return matrix.collect()[0]["pearson({})".format(vector_col)].values[1]
```

```
for cols in numeric_cols:
    print('Correlation {} ~ income: {}'.format(cols,corr(adult_df.select([cols,'income']))))
```

```
Correlation age ~ income: 0.03023705567173963
Correlation industry code ~ income: 0.13769485386944408
Correlation occupation code ~ income: -0.0659489651400274
Correlation wage per hour ~ income: 0.00565547826913644
Correlation capital gains ~ income: 0.2341283178108106
Correlation capital losses ~ income: 0.13855681988295163
Correlation divdends from stocks ~ income: 0.16783592500207165
Correlation detailed household summary in household ~ income: 0.008041787215173388
Correlation family members under 18 ~ income: 0.17174989316039527
Correlation total person income ~ income: 0.012417869564819195
Correlation taxable income amount ~ income: -0.01418193936496789
Correlation hours per week ~ income: 0.21138779808021715
Correlation veterans benefits ~ income: 0.017418923227917424
```

*Hình 5 – Hệ số tương quan giữa biến income với các biến định lượng*

*** Nhận xét:**

Ta thấy hệ số tương quan của biến *income* với các biến *capital gains* và *hours per week* tương đối lớn điều đó chứng tỏ các biến có sự tương quan thuận chặt chẽ với nhau. Mặt khác biến *occupation code* và biến *taxable income amount* có tương quan nghịch với *income*. Trong khi đó biến *wage per hours* và biến*detail household sumary in household* có hệ số tương quan tương đối bé có vẻ như các thông số đó không có nhiều sự ảnh hưởng tới income. Có thể loại bỏ 2 biến trên ra khỏi mô hình.

### *2.2. Tương quan giữa biến income và các biến định tính*

| | sum_sq | df | F | PR(>F) |
|---|---|---|---|---|
| Q("class of worker") | 0.068088 | 8.0 | 0.140966 | 7.073234e-01 |
| education | 498.768697 | 15.0 | 550.733800 | 0.000000e+00 |
| Q("marital status") | 12.107895 | 6.0 | 33.423444 | 1.562466e-40 |
| Q("major industry code") | 0.195753 | 23.0 | 0.140966 | 7.073234e-01 |
| Q("major occupation code") | 0.119154 | 14.0 | 0.140966 | 7.073234e-01 |
| mace | 1.597615 | 4.0 | 6.615244 | 2.557233e-05 |
| Q("hispanic Origin") | 1.534546 | 9.0 | 2.824043 | 2.545138e-03 |
| sex | 84.629025 | 1.0 | 1401.693753 | 3.220962e-305 |
| Q("full or part time employment stat") | 4.702424 | 7.0 | 11.126474 | 3.751269e-14 |
| Q("federal income tax liability") | 9.025746 | 5.0 | 29.898329 | 1.781495e-30 |
| Q("state of previous residence") | 3.764244 | 21.0 | 2.968878 | 5.621509e-06 |
| Q("detailed household and family stat") | 0.244395 | 5.0 | 0.809573 | 5.425474e-01 |
| Q("instance weight") | 0.068088 | 8.0 | 0.140966 | 7.073234e-01 |
| Q("migration code-change in msa") | 0.059577 | 7.0 | 0.140966 | 7.073234e-01 |
| Q("migration code-change in reg") | 0.066783 | 8.0 | 0.138263 | 7.100143e-01 |
| Q("migration code-move within reg") | 0.017022 | 2.0 | 0.140966 | 7.073234e-01 |
| Q("country of birth father") | 4.815298 | 41.0 | 1.945239 | 2.745135e-04 |
| Q("country of birth mother") | 3.916627 | 41.0 | 1.582203 | 1.018847e-02 |
| Q("country of birth self") | 0.348950 | 41.0 | 0.140966 | 7.073234e-01 |
| citizenship | 0.034044 | 4.0 | 0.140966 | 7.073234e-01 |
| Residual | 8245.464924 | 136568.0 | NaN | NaN |

*Hình 6 – Bảng giá trị kiểm định ANOVA các biến định tính với biến income*

*** Nhậnxét:**

Ta thấy giá trị kiểm định các biến *class of worker, education, major industry code, major occupation, detailed household and family stat, instance weight, migration code -change in msa, migration code -change in reg, migration code -move within reg, country of birth self, citizenship*có giá trị P value tương đối lớn ( lớn hơn > 0,5) do đó có thể các biến trên không có nhiều ý nghĩa thống kê trong mô hình. Ta có thể loại bỏ các biến trên khỏi mô hình.

### 3. Lựa chọn đặc trưng cho mô hình đầy đủ

Sau khi phân tích ta lựa chọn các biến cho mô hình đầy đủ bao gồm 26 biến và 136.826 điểm dữ liệu:

```
df_Columns=["wage per hours",
            "detail household sumary in household",
            "class of worker",
            "education",
            "major industry code",
            "major occupation",
            "detailed household and family stat",
            "instance weight",
            "migration code-change in msa",
            "migration code-change in reg",
            "migration code-move within reg",
            "country of birth self, citizenship"
           ]
data = adult_df.drop(*df_Columns)
print("Number of collumns: {}".format(len(data.columns)))
print("Number of row: {}".format(data.count()))
```

```
Number of collumns: 26
Number of row: 136826
```

**4. Lựa chọn mô hình học máy để áp dụng cho bộ dữ liệu**

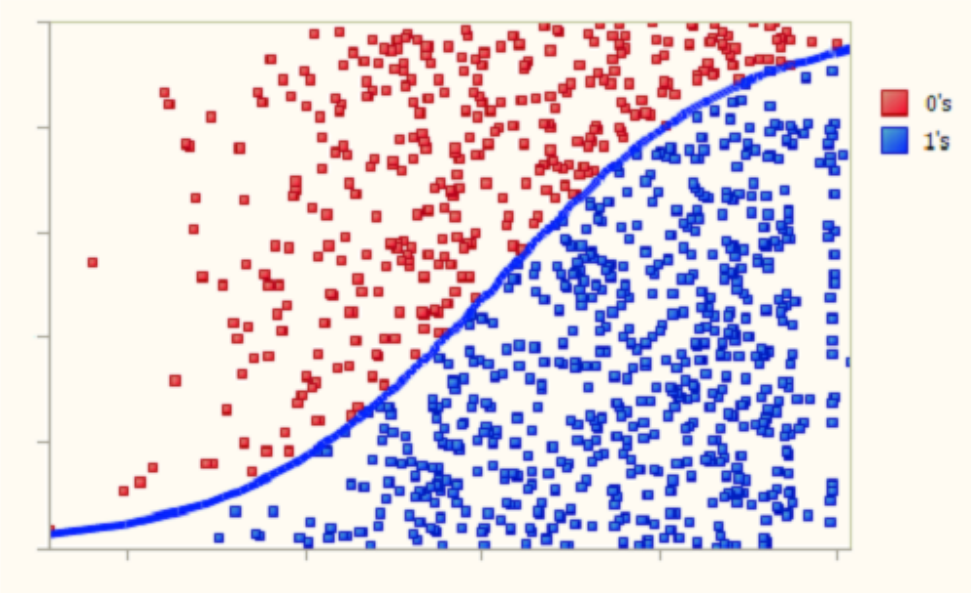
***4.1. Mô hình hồi quy hồi quy logistic (logistic regression)***

Hồi quy logistic là một phương pháp phân tích thống kê được sử dụng để dự đoán giá trị dữ liệu dựa trên các quan sát trước đó của tập dữ liệu.

Mục đích của hồi quy logistic là ước tính xác suất của các sự kiện, bao gồm xác định mối quan hệ giữa các tính năng từ đó đự đoán xác suất của các kết quả, nên đối với hồi quy logisticta sẽ có:

**Input**: dữ liệu input (ta sẽ coi có hai nhãn là 0 và 1).

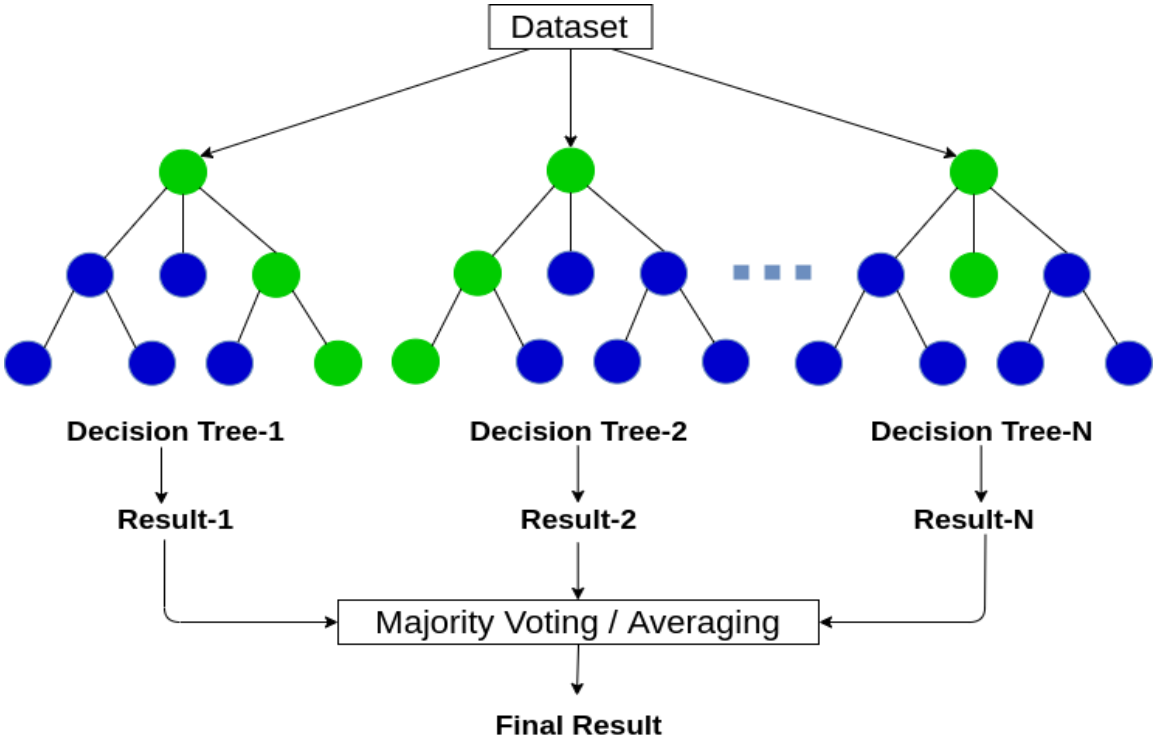
**Output** : Xác suất dữ liệu input rơi vào nhãn 0 hoặc nhãn 1.




*Hình 7 - Mô hình Logistic Regression*

Ở hình trên ta gọi các điểm màu xanh là nhãn 0 và các điểm màu đỏ là nhãn 1 đối với hồi quy logistic ta sẽ biết được với mỗi điểm thì xác xuất rơi vào nhãn 0 là bao nhiêu và xác suất rơi vào nhãn 1 là bao nhiêu, ta có thể thấy giữa hai màu xanh và màu đỏ có một đường thẳng để phân chia rất rõ ràng nhưng nếu các điểm dữ liệu mà không nằm sang hai bên mà nằm trộn lẫn nhiều vào n hau thì ta sẽ phân chia như thế nào? Khi đó ta sẽ gọi tập dữ liệu có nhiều nhiễu và ta phải xử lí trước các nhiễu đó.

### *4.2. Mô hình Random Forest*




*Hình 8 - Mô hình Random forest*

Phương pháp Random Forest sẽ xây dựng nhiều cây quyết định. Tuy nhiên mỗi cây quyết định sẽ khác nha u có yếu tố ngẫu nhiên. Sau đó kết quả dự đoán được tổng hợp từ các cây quyết định. Thực nghiệm đã chỉ ra rằng phương pháp này hiệu quả hơn việc chỉ dùng một decision tree thông thường với khả năng tổng quát hóa tốt hơn, tránh hiện tượng overfit nhưng đánh đổi bằng việc ta không thể hiểu cơ chế hoạt động của thuật toán này do cấu trúc quá phức tạp của mô hình này — do vậy thuật toán này là một trong những phương thức Black Box — tức ta sẽ bỏ tay vào bên trong và rút ra được kết quả chứ không thể giải thích được cơ chế hoạt động của mô hình. Đó là sự đánh đổi giữa khả năng giải thích và khả năng dự báo.

Yếu tố ngẫu nhiên của Random Forest khi sinh ra mỗi cây trong mô hình thể hiện ở hai điểm

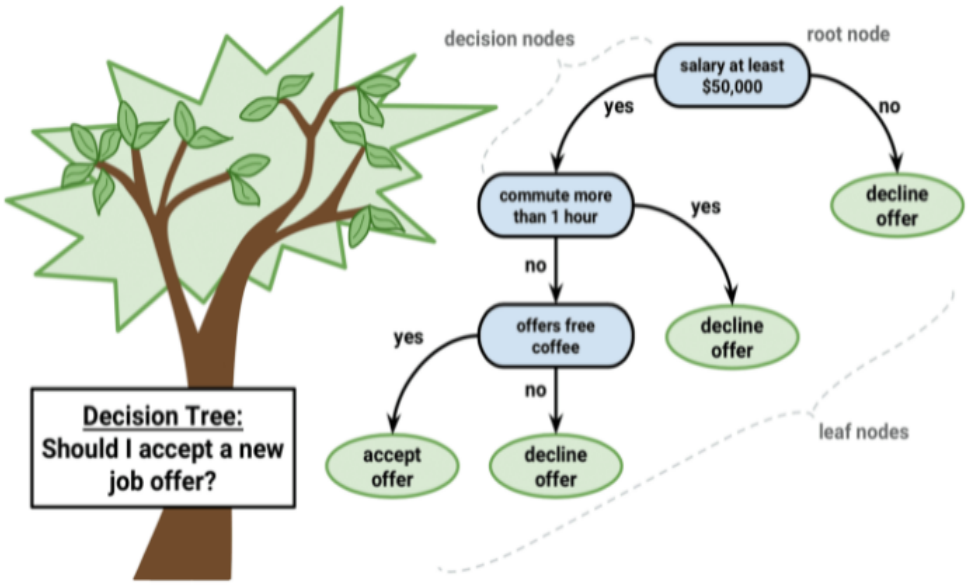
- Mỗi cây quyết định sẽ được học một tập con mẫu khác nhau sinh ngẫu nhiên từ dữ liệu tổng bằng phương pháp boostraping

- Việc lựa chọn feature tại mỗi nhánh của cây sẽ dựa trên feature cho kết quả tốt nhất trong số các feature được lấy ngẫu nhiên thay vì xét trên toàn bộ tất cả cá feature

### *4.3. Mô hình cây quyết định (Decision Tree)*

Decision Tree — một trong những thuật toán phổ biến của Machine Learning thuộc nhánh Supervised Learning. Decision Tree ra đời từ những năm 1975 từ một tác giả có tên Ross Quinlan. Thuật toán này là tiền đề để ra đời những phương pháp dự báo theo d òng Tree-based method như là: Random Forest, Bagging, AdaBoost, Gradient Boosting Machine (GBM) và mới nhất là Extreme Gradient Boosting (XGBoost)— thuật toán đang làm mưa làm gió trong các đấu trường Predictive Modeling Competitive trong khoảng 3 năm trở lại đây.

Một ví dụ đơn giản về Decision Tree như sau: Sau khi đi phỏng vấn vào 1 vị trí công việc. Ứng viên sẽ đối diện với một vài vấn đề như sau: mức lương cho công việc tối thiểu là 50k$ hay ko. Nếu không thì sẽ decline offer. Nếu vượt qua điều kiện về lương ta sẽ xem xét thêm điều kiện tiếp theo là thời gian đi lại đến công ty có quá 1 giờ ko. Nếu quá thì sẽ decline offer. Tuy nhiên nếu ít hơn thì sẽ xem xét điều kiện tiếp theo như có offer cà phê miễn phí vào buổi sáng không. Nếu có thì sẽ accept offe r không thì sẽ decline. Việc quan sát, suy nghĩ và cách thức quyết định của ứng viên sẽ được bắt đầu từ các câu hỏi. Decision tree là một mô hình ra quyết định dựa trên các câu hỏi.




### 5. Áp dụng mô hình vào bộ dữ liệu

Do ta chỉ có 1 bộ dữ liệu du nhất để á p dụng cho mô hình, do vậy ta tiến hành chia tập dữ liệu thành 2 phần : 80% bộ dữ liệu được lấy dùng để training mô hình, 20% dữ liệu còn lại được sử dụng cho quá trình test đánh giá độ chính xác của mô hình:

```
traindataset, testdataset = data.randomSplit((0.8, 0.2))
```

```
traindataset.count()
```

```
109566
```

```
testdataset.count()
```

```
27260
```

Sau khi đã chia bộ dữ liệu ra thành 2 phần ta tiến hành xây dựng pipeline để vector hóa các features sau đó áp dụng các mô hình vào quá trình huấn luyện .

```
numeric_cols
```

```
['age',
 'industry code',
 'occupation code',
 'wage per hour',
 'capital gains',
 'capital losses',
 'divdends from stocks',
 'detailed household summary in household',
 'family members under 18',
 'total person income',
 'taxable income amount',
 'hours per week',
 'veterans benefits']
```

```
string_cols = [i.name for i in data.schema if type(i.dataType) == T.StringType]
len(string_cols), string_cols
```

```
(12,
 ['marital status',
  'major occupation code',
  'mace',
  'hispanic Origin',
  'sex',
  'full or part time employment stat',
  'federal income tax liability',
  'state of previous residence',
  'country of birth father',
  'country of birth mother',
  'country of birth self',
  'citizenship'])
```

```
indexer = StringIndexer(inputCols=string_cols, outputCols=[s + '_Ind' for s in string_cols])
encoder = OneHotEncoder(inputCols=[s + '_Ind' for s in string_cols], outputCols=[s + '_Vector' for s in string_cols])
assembler = VectorAssembler(inputCols=[s + '_Vector' for s in string_cols] + numeric_cols, outputCol="raw_features")
standard_scaler = StandardScaler(inputCol="raw_features", outputCol="features")
```

### *5.1. Mô hình hồi quy logistic regression*

```
logistic = LogisticRegression(labelCol='income',featuresCol='features')
log_pipeline = Pipeline(stages=[indexer, encoder, assembler, standard_scaler, logistic])
```

```
log_model = log_pipeline.fit(traindataset)
log_output = log_model.transform(testdataset)
```

```
logis_prediction = log_output.select('prediction', 'income')
```

```
metrics = ['weightedPrecision', 'weightedRecall', 'f1']
for metric in metrics:
    evaluator = MulticlassClassificationEvaluator(metricName=metric,labelCol = 'income')
    print(metric + ' = ' + str(evaluator.evaluate(logis_prediction)))
```

```
weightedPrecision = 0.917357558722129
weightedRecall = 0.9293103448275862
f1 = 0.9157968333347113
```

*Hình 9 – Áp dụng mô hình hồi quy Logistic regression cho bộ dữ liệu*

*** Nhận xét:**

Ta thấy với mô hình hồi quy logistic regression, hệ số Precision, Recall đều tương đối cao. Cùng với đó chỉ số F1 = 0.916 cho ta thấy độ chính xác của thuật toán là tương đối đáng tin cậy

### *5.2. Mô hình Random forest*

```
random = RandomForestClassifier(labelCol='income',featuresCol='features')
ran_pipeline = Pipeline(stages=[indexer, encoder, assembler, standard_scaler, random])
```

```
ran_model = ran_pipeline.fit(traindataset)
ran_output = ran_model.transform(testdataset)
```

```
ran_prediction = ran_output.select('prediction', 'income')
```

```
metrics = ['weightedPrecision', 'weightedRecall', 'f1']
for metric in metrics:
    evaluator = MulticlassClassificationEvaluator(metricName=metric,labelCol = 'income')
    print(metric + ' = ' + str(evaluator.evaluate(ran_prediction)))
```

```
weightedPrecision = 0.8382389898851795
weightedRecall = 0.915553925165077
f1 = 0.8751922656658621
```

*Hình 10 – Áp dụng mô hình Random forest cho bộ dữ liệu*

*** Nhận xét:**

Ta thấy với mô hình hồi quy logistic regression, hệ số Recall tương đối cao. Tuy nhiên chỉ số F1 = 0.875 cho ta thấy độ chính xác của thuật toán chưa được cao như thuật toán logistic regression.

### *5.3. Mô hình Decision Tree*

```
decision = DecisionTreeClassifier(labelCol='income',featuresCol='features')
dec_pipeline = Pipeline(stages=[indexer, encoder, assembler, standard_scaler, decision])
```

```
dec_model = dec_pipeline.fit(traindataset)
dec_output = dec_model.transform(testdataset)
```

```
dec_prediction = dec_output.select('prediction', 'income')
```

```
metrics = ['weightedPrecision', 'weightedRecall', 'f1']
for metric in metrics:
    evaluator = MulticlassClassificationEvaluator(metricName=metric,labelCol = 'income')
    print(metric + ' = ' + str(evaluator.evaluate(dec_prediction)))
```

```
weightedPrecision = 0.9119381491035213
weightedRecall = 0.9259354365370507
f1 = 0.912703843263812
```

*Hình 11 – Áp dụng mô hình Decision Tree cho bộ dữ liệu*

*** Nhận xét:**

Ta thấy với mô hình hồi quy logistic regression, hệ số Precision, Recall đều tương đối cao. Cùng với đó chỉ số F1 = 0.913 cho ta thấy độ chính xác của thuật toán là tương đối đáng tin cậy.

**6. Kết luận**

Việc thu thập và xử lý dữ liệu là một trong những khó khăn lớn nhất trong dự đoán mức thu nhập người trưởng thành . Nghiên cứu này chú trọng vào việc phân tích, xử lý dữ liệu bị thiếu (missing), bị trùng lặp, dữ liệu ngoại lệ, đánh giá tương quan giữa các b iến giải thích để tạo ra bộ dữ liệu chuẩn (clearning data) trước khi áp dụng vào mô hình dự đoán.

Trong quá trình thực hiện, nghiên cứu này tập trung vào việc xem xét vai trò, đánh giá tầm quan trọng của các biến giải thích và chọn ra mô hình được xem là tốt nhất. Chọn mô hình tối ưu hay chọn biến số liên quan có ý nghĩa quan trọng vì dữ liệu lớn, phức tạp, số lượng biến giải thích lớn khiến cho quá trình phân tích thực sự là 1 thử thách. Khi phát hiện và chọn ra được các yếu tố liên quan sẽ giúp khám phá tốt hơn và cho ra kết quả tin cậy hơn. Kết quả này có nghĩa là các yếu tố ảnh hưởng được chọn có thể giải thích hầu hết các yếu tố quyết định mức thu nhập của người trưởng thành .

Nhìn chung, đóng góp của nghiên cứu này nằm ở phương pháp nghiên cứu sử dụng kết hợp nhiều phương pháp phân tích, xử lý dữ liệu và đánh giá các chỉ số trong mô hình, tạo điều kiện cho việc dự đoán mức thu nhập của người trưởng thành dự trên dữ liệu điều tr a dân số của Cục Điều tra dân số Hoa Kỳ.

**TÀI LIỆU THAM KHẢO**

1. Bài giảng, tài liệu của giảng viên môn Mining Big Datasets.
2. Tài liệu về điều tra dân số: Hồ sơ nhân khẩu học hàng năm, 1994 (ICPSR 646I), phiên bản 16/3/1995.
   https://www.icpsr.umich.edu/web/ICPSR/studies/6461/summary
3.